Claims

We claim:

1. A computer-implemented method for implementing a synchronous reactive system in a graphical program, the method comprising:

including a loop structure in the graphical program in response to first user input, wherein the loop structure is operable to execute iteratively in a synchronous manner, and wherein the loop structure is operable to store state information for a current iteration and to provide the state information for use in a next iteration of the loop; and

including a plurality of graphical program nodes in the loop structure in response to second user input;

wherein, during execution of the graphical program, for each iteration, the loop structure is operable to execute the plurality of graphical program nodes in the loop structure using stored state information from an immediately previous iteration, wherein the loop structure performs each iteration subject to a time constraint, and wherein the loop structure executes the plurality of graphical program nodes in a synchronous reactive manner.

2. The method of claim 1,

wherein the loop structure comprises at least one synchronization register, comprising an input and an output; and

wherein, in said loop structure being operable to store and provide the state information, the input to the synchronization register is operable to store the state information, and the output of the synchronization register is operable to provide the state information.

3. The method of claim 2, further comprising:

executing the graphical program, wherein said executing comprises:

executing the loop structure in an iterative manner until a stopping condition obtains.

4. The method of claim 3, wherein said executing the loop structure comprises:

for each iteration,

the output of the synchronization register providing the state information for the immediately previous iteration for use in the current iteration; and

the input to the synchronization register storing the state information for the current iteration for use in the next iteration.

5. The method of claim 4,

wherein the loop structure corresponds to a synchronous reactive hardware circuit; and

wherein a single bit synchronization register corresponds to a flip flop comprised in the synchronous reactive hardware circuit, and wherein a multi-bit synchronization register corresponds to a bank of flip flops comprised in the synchronous reactive hardware circuit.

6. The method of claim 4, wherein the loop structure comprises combinatorial logic which calculates the state information to be stored in the synchronization register to be used for the next loop iteration.

7. The method of claim 1, wherein the at least one synchronization register is implemented in one or more of:

hardware; and

software.

8. The method of claim 1, wherein the loop structure comprises a single-cycle timed loop, and wherein in being operable to execute in a synchronous manner, the loop structure is operable to iterate each clock cycle.

9. The method of claim 1, wherein the loop structure comprises an N-cycle loop, and wherein in being operable to execute in a synchronous manner, the loop structure is operable to iterate every N clock cycles, wherein N is greater than one.

10. The method of claim 1, wherein said including the plurality of graphical program nodes in the loop structure comprises pipelining at least a subset of the plurality of graphical program nodes in the loop structure, thereby generating a pipeline inside the loop structure, wherein the pipeline has an associated pipeline depth, resulting in a corresponding latency between each input of the loop structure and corresponding output of the loop structure.

11. The method of claim 1, the method further comprising:

including one or more additional loop structures in the graphical program in response to third user input, wherein the one or more additional loop structures are each operable to execute iteratively in a synchronous manner, and wherein the one or more additional loop structures are each operable to store state information for a current iteration and to provide the state information for use in a next iteration of the loop; and

including a respective plurality of graphical program nodes in each of the one or more additional loop structures in response to fourth user input;

wherein, during execution of the graphical program, for each iteration, each of the one or more additional loop structures is operable to execute the respective plurality of graphical program nodes in the loop structure using stored state information from an immediately previous iteration, and wherein each of the one or more additional loop structures performs each iteration subject to a respective time constraint.

12. The method of claim 11, further comprising:

receiving user input specifying a respective timing source for each of the loop structure and each of the one or more additional loop structures in the graphical program; and

associating each respective timing source with each of the loop structure and each of the one or more additional loop structures, wherein each of the loop structure and each of the one or more additional loop structures is then operable to execute iteratively in accordance with the associated timing source.

13. The method of claim 12, wherein each of the timing sources comprises a respective clock with a respective frequency.

14. The method of claim 11, wherein at least a subset of the one or more additional loop structures comprise nested loops.

15. The method of claim 1, wherein, in being operable to execute in a synchronous manner, the loop structure is operable to receive an input and to provide an output each iteration.

16. The method of claim 15, wherein in being operable to receive an input and to provide an output each iteration, the loop structure is operable to receive the input and to provide the output on a single clock edge.

17. The method of claim 1, further comprising:

receiving input specifying a first clock for the loop structure; and

configuring the loop structure to receive clock signals from the specified first clock, wherein the loop structure is then operable to execute synchronously in accordance with the specified first clock.

18. The method of claim 17, wherein the graphical program includes a second plurality of graphical program nodes configured to execute in accordance with a second clock, wherein the second clock has a different frequency from that of the first clock.

19. The method of claim 1, the method further comprising:

configuring the loop structure to iterate every N cycles, wherein N is greater or equal to 1;

analyzing the graphical program to determine whether the plurality of graphical program nodes in the loop structure are executable in N cycles; and

if the plurality of graphical program nodes in the loop structure are not executable in N cycles, outputting an error message.

20. The method of claim 19, wherein said analyzing the graphical program is performed at edit time of the graphical program.

21. The method of claim 1,

wherein the loop structure and the plurality of graphical program nodes implements a state machine comprising a plurality of states and a plurality of state transitions;

wherein transitions between states are evaluated every iteration of the loop structure; and

wherein a current state of the state machine is stored in a synchronization register comprised in the loop structure.

22. The method of claim 1, further comprising:

displaying a graphical user interface (GUI) for creating and configuring the graphical program, wherein said first user input and said second user input are received to the GUI.

23. The method of claim 1,

wherein the graphical program comprises a plurality of interconnected nodes that visually indicate functionality of the graphical program.

24. The method of claim 1, wherein said second user input comprises second user input arranging the plurality of nodes on a display, and interconnecting the plurality of nodes.

25. The method of claim 1,

wherein the graphical program comprises a block diagram portion and a user interface portion.

26. The method of claim 25,

wherein, during execution of the graphical program, the user interface is displayed on a display of a first computer system and the block diagram executes on a second computer system.

27. The method of claim 1,

wherein the graphical program comprises a graphical data flow program.

28. The method of claim 1,

wherein the graphical program is operable to perform one or more of:

an industrial automation function;

a process control function;

a hardware circuit function;

a test and measurement function; and

a simulation function.

29. The method of claim 1, the method further comprising:

deploying the graphical program to a hardware device for execution.

30. The method of claim 29, wherein the hardware device comprises a programmable hardware element, and wherein said deploying the graphical program to a hardware device for execution comprises:

compiling the graphical program to generate a hardware configuration program; and

deploying the hardware configuration program to the programmable hardware element.

31. A computer-implemented method for implementing a reactive system in a graphical program, the method comprising:

receiving first user input specifying a loop structure for inclusion in the graphical program, wherein the loop structure is operable to execute iteratively, and wherein the loop structure is operable to store state information for a current iteration and to provide the state information for use in a next iteration of the loop;

including the loop structure in the graphical program in response to the first user input;

receiving second user input specifying a plurality of graphical program nodes for inclusion in the graphical program; and

including the plurality of graphical program nodes in the loop structure in response to the second user input;

wherein, during execution of the graphical program, for each iteration, the loop structure is operable to execute the plurality of graphical program nodes in the loop structure using stored state information from an immediately previous iteration, wherein the loop structure performs each iteration subject to a time constraint, and wherein the loop structure executes the plurality of graphical program nodes in a synchronous reactive manner.

32. A system for implementing a synchronous reactive system in a graphical program, the system comprising:

a processor; and

a memory medium, coupled to the processor, wherein the memory medium stores program instructions executable by the processor to:

include a loop structure in the graphical program in response to first user input, wherein the loop structure is operable to execute iteratively in a synchronous manner, and wherein the loop structure is operable to store state information for a current iteration and to provide the state information for use in a next iteration of the loop; and

include a plurality of graphical program nodes in the loop structure in response to second user input;

wherein, during execution of the graphical program, for each iteration, the loop structure is operable to execute the plurality of graphical program nodes in the loop structure using stored state information from an immediately previous iteration, wherein the loop structure performs each iteration subject to a time constraint, and wherein the loop structure executes the plurality of graphical program nodes in a synchronous reactive manner.

33. A computer-implemented system for implementing a synchronous reactive system in a graphical program, the system comprising:

means for including a loop structure in the graphical program in response to first user input, wherein the loop structure is operable to execute iteratively in a synchronous manner, and wherein the loop structure is operable to store state information for a current iteration and to provide the state information for use in a next iteration of the loop; and

means for including a plurality of graphical program nodes in the loop structure in response to second user input;

wherein, during execution of the graphical program, for each iteration, the loop structure is operable to execute the plurality of graphical program nodes in the loop structure using stored state information from an immediately previous iteration, wherein the loop structure performs each iteration subject to a time constraint, and wherein the loop structure executes the plurality of graphical program nodes in a synchronous reactive manner.